

**HOCHSCHULE
HANNOVER**
UNIVERSITY OF
APPLIED SCIENCES
AND ARTS
–
*Fakultät IV
Wirtschaft und
Informatik*

Effiziente Lichtsimulation in der Computergrafik: Ein Vergleich von Ray Tracing und Photon Mapping

Tobias Selzer

Seminar-Arbeit im Studiengang „Angewandte Informatik“

17. Juli 2024



Autor 1: Tobias Selzer
Matrikelnummer: 1766567
tobias.selzer@stud.hs-hannover.de

Prüferin: Prof. Dr. Ingo Ginkel
Abteilung Informatik, Fakultät IV
Hochschule Hannover
ingo.ginkel@hs-hannover.de

Selbständigkeitserklärung

Mit der Abgabe der Ausarbeitung erklären wir, dass wir die eingereichte Seminar-Arbeit selbständig und ohne fremde Hilfe verfasst, andere als die von uns angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht haben.

Tobias Selzer

Hannover, den 17. Juli 2024

Inhaltsverzeichnis

1	Einleitung	5
1.1	Hintergrund und Motivation	5
1.2	Aufbau der Arbeit	5
2	Hintergrund Mathematik	6
2.1	Rendering Equation	6
2.2	BRDF	7
3	Raycasting & Ray Tracing	8
3.1	Exkurs: Lokale Beleuchtungsmodelle	8
3.2	Raycasting	8
3.2.1	Wie funktioniert Raycasting?	9
3.3	(Eye) Ray Tracing nach J. Turner Whitted	9
3.4	Eye Ray Tracing vs. Light Ray Tracing	11
3.5	Weiterentwicklungen	11
3.6	Exkurs: Radiosity	12
4	Photon Mapping	14
4.1	Was ist eine Photon Map?	14
4.2	Durchgang 1: Generieren der Photon Maps	15
4.3	Durchgang 2: Rendering	16
4.3.1	Term 1: Direkte Beleuchtung	17
4.3.2	Term 2: Spiegelnde Reflexionen	17
4.3.3	Term 3: Caustics	18
4.3.4	Term 4: Weiches indirektes Licht	18
5	Unterschiede & Zusammenhänge von Ray Tracing & Photon Mapping	19
5.1	Grundprinzip	19
5.2	Beleuchtungseffekte	19
5.3	Einschränkungen	19
5.4	Anwendungsgebiete	19
6	Effekte von Raytracing und Photon Mapping	21

7	Entwicklungen und Zukunftsaussichten für Ray Tracing und Photon Mapping	25
7.1	Allgemeine Entwicklungen	25
7.2	Ray Tracing	25
7.3	Photon Mapping	26
	Literaturverzeichnis	27

1 Einleitung

Im Rahmen dieser Arbeit wird das Ray Tracing-Verfahren und die Erweiterung durch Photon Mapping behandelt.

In diesem Kapitel wird zunächst der Hintergrund und die Motivation beleuchtet (**Abschnitt 1.1**). Das Kapitel schließt mit einem Überblick über die Arbeit und die behandelten Themen (**Abschnitt 1.2**).

1.1 Hintergrund und Motivation

Ray Tracing ist ein Algorithmus der Computergrafik, welcher es ermöglicht realistische Lichtsimulationen zu erstellen. Erstmals 1963 eingesetzt [Mas07], gilt Ray Tracing mittlerweile als eines der am meisten geschätzten und leistungsfähigsten Verfahren für das Rendern von Bildern. Erweitert wird Ray Tracing durch Photon Mapping, welches einzelne Funktionen mit sich bringt, die nicht oder nur schwer mit Ray Tracing abzubilden wären. Die Motivation für diese Seminararbeit liegt in der zunehmenden Bedeutung realitätsnaher Grafiken in Bereichen wie Film, Spieleentwicklung und virtueller Realität. Fortschritte in der Hardware und Algorithmenoptimierung haben diese Verfahren zugänglicher gemacht, was ihr Potential für Echtzeitanwendungen erhöht. [Gri08] [Gla19]

1.2 Aufbau der Arbeit

Es folgt ein Überblick über den Aufbau dieser Arbeit. Beginnend mit, **Kapitel 2**, der grundlegenden Mathematik die in den Folgekapiteln angewendet wird. **Kapitel 3** setzt sich mit Raycasting und Ray Tracing auseinander. **Kapitel 4** beschäftigt sich mit Photon Mapping. In **Kapitel 5** werden die Unterschiede und die Zusammenhänge von Ray Tracing und Photon Mapping beleuchtet. **Kapitel 6** umfasst Effekte, die mit Hilfe von Ray Tracing und Photon Mapping erzeugt werden können. Kapitel **Kapitel 7** schließt mit Entwicklungen und Zukunftsaussichten von Ray Tracing und Photon Mapping.

2 Hintergrund Mathematik

2.1 Rendering Equation

Jedes physikalisch korrekte Beleuchtungsmodell basiert auf einer Rendering Equation (Rendergleichung) und versucht diese für die Lichtberechnungen zu lösen (Ray Tracing, Photon Mapping, Path Tracing etc.). Die Gleichung vereint alle Aspekte eines physikalisch akkuraten Modells in einer mathematischen Formel. Im Jahr 1986 stellte James T. Kajiya erstmals die Idee vor verschiedene Rendering-Algorithmen in einer einzigen Gleichung zusammenzufassen. Die aufgestellte Rendering Equation sieht wie folgt aus (Formel 2.1):

$$I(x, x') = g(x, x') \left[\epsilon(x, x') + \int_S \rho(x, x', x'') I(x', x'') dx'' \right] \quad (2.1)$$

Photon Mapping verwendet die gleiche Rendering Equation nur mit anderen Bezeichnungen, weshalb diese anstelle der ursprünglichen von Kajiya entwickelten Gleichung näher beschrieben wird (Formel 2.2):

$$L_s(x, \Psi_r) = L_e(x, \Psi_r) + \int_{\Omega} f_r(x, \Psi_i, \Psi_r) L_i(x_i, \Psi_i) \cos \theta_i d\omega_i \quad (2.2)$$

$L_s(x, \Psi_r)$	Ausgehende Strahlungsdichte von Punkt x in Richtung Ψ_r (Farbe des Pixels)
$L_e(x, \Psi_r)$	Emittierte Strahlungsdichte (der Oberfläche) an Punkt x in Richtung Ψ_r
$f_r(x, \Psi_i, \Psi_r)$	BRDF (der Oberfläche), beschreibt das Reflexionsverhalten an Punkt x von Ψ_i in Richtung Ψ_r
$L_i(x, \Psi_i)$	Einfallende Strahlungsdichte an Punkt x aus Richtung Ψ_i
$\cos \theta_i$	Kosinus des Einfallswinkels zwischen Ψ_i und der Oberflächennormalen n
$\int_{\Omega} d\omega_i$	Integral über alle einfallenden Richtungen Ψ_i

Bei Whitted-style Ray Tracing hingegen kommt eine andere Rendering Equation zum Einsatz, welche die Berechnungen des Phong-Beleuchtungsmodells erweitert (Formel 2.3).

$$I = I_a + k_d \sum_{j=1}^{j=ls} (\bar{\mathbf{N}} \cdot \bar{\mathbf{L}}_j) + k_s S + k_T T \quad (2.3)$$

I	Gesamt Intensität (Farbe des Pixels)
I_a	Intensität des Umgebungslichts (Ambient Light)
k_d	Diffuse Reflexionskonstante
$\sum_{j=1}^{j=ls}$	Summe über alle Lichtquellen j
$\bar{\mathbf{N}}$	Normalenvektor an der Oberfläche
$\bar{\mathbf{L}}_j$	Vektor zur Lichtquelle j
k_s	Spekulare Reflexionskonstante (Reflection)
S	Intensität des spekularen Lichtanteils (Reflection)

k_T	Transmissionskonstante (Refraction)
T	Intensität des Transmissionslichtanteils (Refraction)

Zusammenfassend kann festgehalten werden, dass das Resultat der Rendering Equations die ausgehende Strahlungsdichte (Radiance) beziehungsweise die Intensität repräsentiert, welche beim Photon Mapping beziehungsweise Ray Tracing der Farbe des Pixels im gerenderten Bild entspricht.

Des Weiteren ist zu erwähnen, dass Teile der Rendering Equations rekursiv berechnet werden. Details hierzu finden sich in Kapitel 3 beziehungsweise Kapitel 4. [Jen] [Kaj] [HV] [Whi]

2.2 BRDF

Die Bidirectional Reflectance-Distribution Function $f_r(x, \Psi_i, \Psi_r)$ (kurz BRDF) ist Teil der in Abschnitt 2.1 erläuterten Renderinggleichung. Die Verteilungsfunktion gibt an, wie viel Licht von der Einfallrichtung (Ψ_i) zur Reflexionsrichtung (Ψ_r) an einem bestimmten Punkt (x) auf einer Oberfläche reflektiert wird. Sie wird verwendet, um das Reflexionsverhalten verschiedener Materialien zu modellieren. In den meisten Echtzeit Rendering Verfahren wird die Cook-Torrance-BRDF eingesetzt. [Nic+77] [Moh23]

3 Raycasting & Ray Tracing

3.1 Exkurs: Lokale Beleuchtungsmodelle

Um den wesentlichen Unterschied von Raycasting beziehungsweise Ray Tracing zu verstehen ist es notwendig zuerst auf lokale Beleuchtungsmodelle zu schauen. Das wohl bekannteste lokale Beleuchtungsmodell ist das Phong-Beleuchtungsmodell. Bei diesem werden die Berechnungen für die Farbwerte der Pixel pro Objekt durchgeführt (lokal berechnet). Dies hat zur Folge das nur die direkte Beleuchtung (Licht in gerader Linie von den Lichtquellen der Szene) Einfluss auf die Pixelfarbe hat. Indirekte Beleuchtung zwischen Objekten sowie die Sichtbarkeit der Lichtquelle und die damit verbundenen Schatten werden nicht ausreichend berücksichtigt (nur durch den ambienten Anteil simuliert). Folglich ist das Phong-Beleuchtungsmodell aus physikalischer Sicht inkorrekt. Die Farbe eines Pixels bildet die folgende Summe (Formel 3.1, Bild 3.1).

$$\begin{aligned} \text{Pixelfarbe} = & \\ & \text{Objekt-Farbe} * \text{Umgebungslichtfarbe (Ambient)} + \\ & \text{Streuung der Lichtquelle auf dem Objekt (Diffuse)} + \\ & \text{Reflektion der Lichtquelle auf dem Objekt (Specular)} \\ & (3.1) \end{aligned}$$

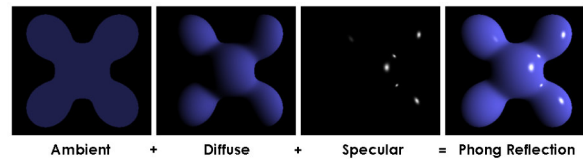


Abbildung 3.1: Bildliche Darstellung der einzelnen Summenanteile, aus denen sich die Pixelfarbe ergibt (Quelle: [Wik04])

An dieser Stelle sei angemerkt, dass der Specular-Anteil der Berechnung in Abhängigkeit von der Position des Beobachters variiert. Des Weiteren ist festzuhalten, dass zusätzlich zur Berechnung der Pixelfarbe ein weiteres Verfahren notwendig ist, um die vom Phong-Beleuchtungsmodell vernachlässigten Verdeckungen (das Verdeckungsproblem) zu berücksichtigen. Als Verfahren kann beispielsweise Z-Buffering eingesetzt werden. [Moh23] [LGK] [UniFre]

3.2 Raycasting

Raycasting ist das Verfahren, welches Ray Tracing zugrunde liegt. Es handelt sich um ein globales Beleuchtungsmodell, welches bei der Berechnung der Pixelfarbe auch Verdeckungen des Lichts durch benachbarte Objekte mit einbezieht. Damit ist Raycasting im Gegensatz zum Phong-Beleuchtungsmodell ein physikalisch korrektes Beleuchtungsmodell. Allerdings findet auch hier das indirekte Licht keine ausreichende Berücksichtigung, da es ebenfalls nur durch den ambienten Anteil simuliert wird.

3.2.1 Wie funktioniert Raycasting?

Zunächst wird der Bildschirm (beziehungsweise die Near Plane) gerastert (in ein Raster von Pixeln aufgeteilt). Anschließend wird durch jedes Pixel vom Eye (Augpunkt) ausgehend ein Ray (Strahl) in die 3D-Szene gesendet. Dieser Ray wird verfolgt, bis er auf ein Objekt trifft oder eine maximale Tiefe erreicht wurde. Befinden sich mehrere Objekte in der Richtung des gesendeten Rays so wird das zuerst getroffene Objekt für die Berechnung genutzt. Raycasting löst damit implizit das Verdeckungsproblem. Am Schnittpunkt wird dann das lokale Beleuchtungsmodell (Phong) berechnet. Zusätzlich werden sogenannte Shadow Rays (Schattenstrahlen) vom Schnittpunkt aus zu allen Lichtquellen gesendet (Bild 3.2). Wird auch hier ein Objekt getroffen liegt der Pixel im Schatten des getroffenen Objektes und hat zur Folge das für das lokale Beleuchtungsmodell nur der ambiente Anteil berücksichtigt wird (die Pixelfarbe ist also entsprechend dunkler) (Bild 3.3). Ist der Weg zur Lichtquelle frei wird das Pixel direkt beleuchtet (direkter Lichtanteil). Die endgültige Pixelfarbe ergibt sich durch die Summierung der Beleuchtungskomponenten. [Uni] [ESK97] [Whi18] [The]

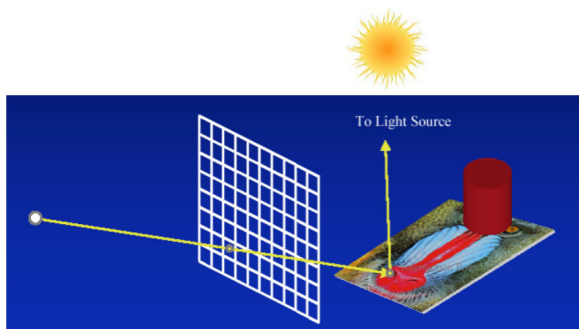


Abbildung 3.2: Shadow Rays vom Schnittpunkt zu allen Lichtquellen senden (Quelle: [Uni])

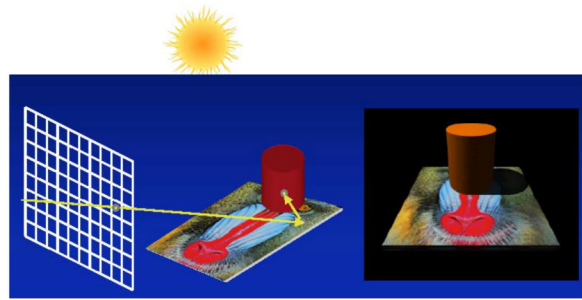


Abbildung 3.3: Treffen die Schattenstrahlen auf ein Objekt, so liegt der Punkt im Schatten (Quelle: [Uni])

3.3 (Eye) Ray Tracing nach J. Turner Whitted

Ray Tracing ist ein globales Beleuchtungsmodell. Wie bereits in Abschnitt 3.2 erwähnt basiert es im wesentlichen auf Raycasting. Der Berechnungsablauf ist derselbe wie beim Raycasting mit dem wesentlichen Unterschied, dass das spiegelnde indirekte Licht (Specular) berücksichtigt wird, das diffuse indirekte Licht wird allerdings auch bei Ray Tracing nicht berücksichtigt. Für das indirekte Licht werden dazu zwei weitere Rays vom Schnittpunkt aus ausgesandt. Zum einen ist das ein Ray für das ideal reflektierte Licht (Reflection **R**). Dieser Ray muss mindestens über ein weiteres Objekt reflektieren und eine Lichtquelle treffen damit reflektiertes Licht mit in die Berechnung einfließt. Zum anderen ist das ein Ray für das ideal gebrochene Licht (Refraction **T**). Dieses Licht, welches der Ray verfolgt, transmittiert durch das Objekt mit dem Schnittpunkt (Bild 3.4). In der Realität treten diese Effekte bei stark spiegelnden Oberflächen auf (Reflection), beispielsweise bei Spiegeln oder bei lichtdurchlässigen Oberflächen (Refraction), wie beispielsweise bei Glas (Bild 6.2).

Die endgültige Pixelfarbe (I : Intensität) ergibt sich aus der Summe der lokalen Beleuchtungskomponenten (I_*), der gewichteten Leuchtdichten der ideal reflektierten Rays ($I_{\text{reflection}}$) und der gewichteten Leuchtdichten der ideal gebrochenen Rays ($I_{\text{refraction}}$) (Formel 3.2) (Bild 3.4). Die Formel wurde der Übersicht halber vereinfacht und entspricht der Rendering Equation aus Abschnitt 2.1.

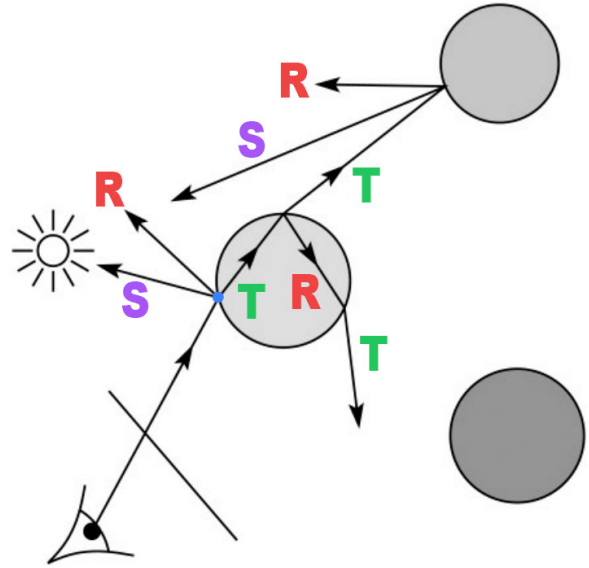


Abbildung 3.4: [Angepasstes] Whitted-style Ray Tracing; Lokales Beleuchtungsmodell (Schnittpunkt), Shadow Ray (S), Reflection (R), Refraction (T) (Quelle: [The])

$$I = I_{\text{ambient}} + (\text{Shadow-factor} \cdot (I_{\text{diffuse}} + I_{\text{specular}})) + I_{\text{reflection}} + I_{\text{refraction}} \quad (3.2)$$

Der Algorithmus wird aufgrund der Rekursion teilweise auch als zweistufig beschrieben. Zunächst wird der Baum mit Hilfe der Schnittpunktberechnungen aufgebaut und anschließend traversiert. [Uni] [ESK97] [Whi18] [The]

Die Formel zeigt die Verwendung eines Schattenfaktors (Shadow-factor), dieser ist bereits aus dem Raycasting bekannt und dient zur Anpassung der Formel, wenn der Schattenstrahl durch ein Objekt blockiert wird. Weiterhin ist zu beachten, dass der Algorithmus einen Strahlbaum mit definierter Rekursionstiefe (idealerweise 5) verwendet, da die Leuchtdichten des reflektierten und des gebrochenen Lichts rekursiv berechnet werden.

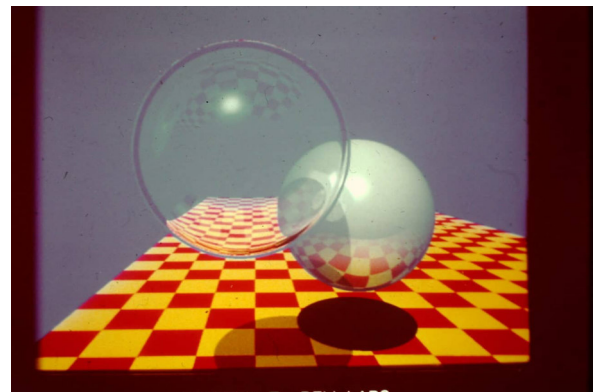


Abbildung 3.5: Erstes Whitted-style Ray Tracing Bild (Quelle: [Whi18])

3.4 Eye Ray Tracing vs. Light Ray Tracing

Im vorherigen Abschnitt 3.3 wurde das Ray Tracing nach J. Turner Whitted (Whitted-style ray tracing / Ray Tracing) behandelt. Dieser Algorithmus zählt zum Typ des Eye Ray Tracings. Eye Ray Tracing wird mitunter auch als Backward Ray Tracing bezeichnet. Dies liegt daran, dass hier die Rays vom Eye aus in die Szene gesendet werden (Bild 3.6). Alternativ können Rays auch vom Light (den Lichtquellen) aus in Szene gesendet werden, dann wird von Light-Ray Tracing bzw. Forward Ray Tracing oder Photon Tracing gesprochen (Bild 3.7). [Lev]

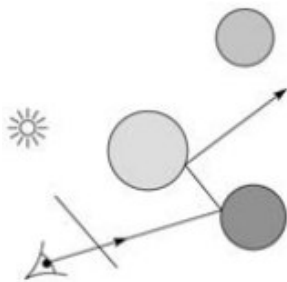


Abbildung 3.6: Eye Ray Tracing (Quelle: [Lev])

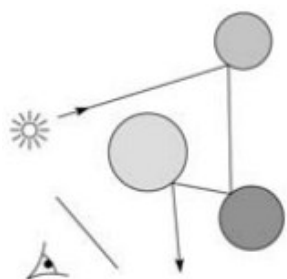


Abbildung 3.7: Light Ray Tracing (Quelle: [Lev])

3.5 Weiterentwicklungen

Im Folgenden wird kurz auf Weiterentwicklungen zu den bisher vorgestellten Algorithmen eingegangen (Bild 3.8 zeigt den Zusammenhang beziehungsweise die Hierarchie). Mit *Distribution Ray Tracing* existiert eine Verfeinerung des bekannten Ray Tracing Verfahrens. Hier werden mehrere Rays zur Bestimmung des endgültigen Pixelwertes in Form des Durchschnitts aus einer Reihe von Stichprobenschätzungen verwendet, um den endgültigen Pixelwert zu bestimmen. Diese Erweiterung kann je nach Anzahl der Rays (auch Samples genannt) Aliasing verringern beziehungsweise verhindern. Außerdem können (besser) verschwommene Reflexionen, Brechungen, weiche Schatten, Tiefenunschärfe und Bewegungsunschärfe dargestellt werden.

Als *Monte Carlo Verfahren* werden konkrete Implementierungen von Distribution Ray Tracing bezeichnet. Hierbei wird die Monte Carlo Integration (Monte Carlo Simulation) - ein Stochastik-Verfahren, bei dem wiederholt Zufallsstichproben gezogen werden - genutzt, um die Rendergleichung näherungsweise zu bestimmen.

Es existieren unter anderem *Monte Carlo Ray Tracing* (auch *Path Tracing* genannt), *Bidirectional Path Tracing* und *Metropolis Light Transport* als Verfahren, die auf dem Monte Carlo Verfahren aufbauen.

Photon Mapping und dessen verbesserte Version, das Progressive Photon Mapping, die in Kapitel 3 behandelt werden, nutzen Monte Carlo Ray Tracing für ihre Berechnungen. [Uni] [HOJ] [Jen] [Kro+] [Mar]

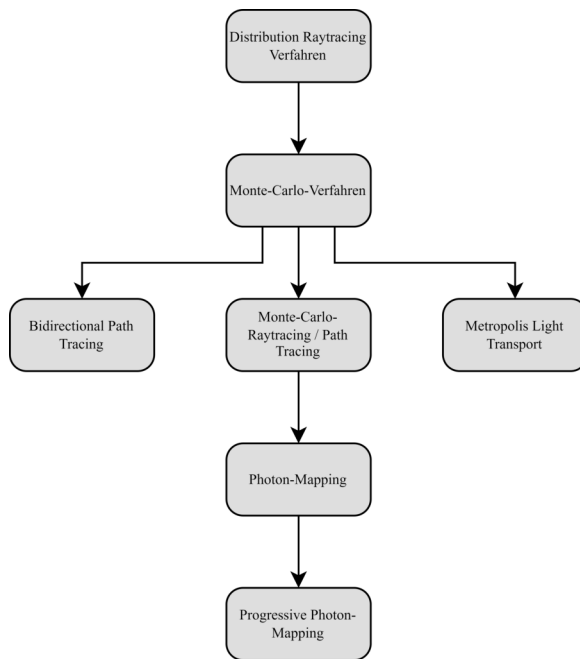


Abbildung 3.8: Zusammenhang beziehungsweise Hierarchie der Weiterentwicklungen von Ray Tracing (Quelle: Entwickelt aus [Uni] [HOJ] [Jen] [Kro+] [Mar])

Wert ist für nicht leuchtende Patches anfangs null. Nach der Erstellung der Patches beginnt ein iterativer Berechnungsprozess, bei dem wiederholt berechnet wird, wie viel Lichtenergie von jedem Patch zu allen anderen Patches übertragen wird. Dieser Prozess läuft so lange, bis das System konvergiert, das heißt bis die Leuchtdichten stabil und unverändert bleiben.

Die Berechnung der Energieübertragung zwischen den Patches basiert auf sogenannten Formfaktoren, die den Anteil der Lichtenergie darstellen, der von einem Patch zu einem anderen übertragen wird. Diese Faktoren berücksichtigen die folgenden Eigenschaften der Patches: die relative Orientierung, die Form, den Abstand zwischen den Patches, sowie Verdeckungen durch andere Patches.

Die grundlegende Radiosity-Gleichung für normierte Einheitsflächen von lambertschen diffusen Patches lautet (Formel 3.3):

$$B_i = E_i + \rho_i \sum_{1 \leq j \leq n} \frac{(B_j A_j) \cdot F_{j-i}}{A_i} \quad (3.3)$$

3.6 Exkurs: Radiosity

Radiosity stellt ein auf einem thermischen Modell basierendes Verfahren dar, welches die Lichtenergieverteilung in einer Szene in einer realistischen Weise simuliert. Im Gegensatz zu Ray Tracing, welches den Fokus auf das spiegelnde indirekte Licht legt, behandelt Radiosity die diffusen Interaktionen zwischen den Oberflächen in einer Szene.

Radiosity beginnt mit der Unterteilung der Szene in kleinere Flächenstücke, sogenannte "Patches". Jede dieser Patches kann Lichtenergie emittieren und reflektieren. Dazu startet jeder Patch mit einem initialen Leuchtdichtewert (Energiewert). Dieser

B_i	Totale Strahlungsdichte in Watt/m ²
E_i	Emittiertes Licht in Watt/m ²
ρ_i	Anteil der einfallenden Energie, die vom i-ten Patch reflektiert wird
$B_j A_j$	Gesamte von Patch j abgestrahlte Energie mit Fläche A_j
F_{j-i}	Anteil der Energie, die das j-te Patch verlässt und in Patch i ankommt
A_i	Fläche des i-ten Patches

Zu den Vorteilen von Radiosity zählt die realistische Beleuchtung diffuser Oberflächen, einschließlich Lichtwechselwirkungen wie Color Bleeding und weicher Schatten.

Weiterhin ist es möglich, die Szene aus verschiedenen Blickwinkeln zu rendern, ohne dass eine erneute Berechnung der Leuchtdichten erforderlich wird.

Allerdings sind mit Radiosity auch Einschränkungen verbunden. Die Berechnungen sind sehr aufwendig, da sie die Wechselwirkungen zwischen allen Patches berücksichtigen müssen. Zudem geht Radiosity von ideal diffusen Reflexionen aus und berücksichtigt keine spekulären Reflexionen oder Lichtbrechungen.

Um sowohl diffuse als auch spiegelnde Lichtwirkungen realistisch darstellen zu können, wird Radiosity häufig mit anderen Methoden wie Ray Tracing kombiniert. [\[Hv\]](#) [\[Jen\]](#)

4 Photon Mapping

4.1 Was ist eine Photon Map?

Photon Maps lassen sich als balancierte k-d-Bäume zur Speicherung von Photonen beschreiben. Im zweiten Durchgang werden sie dazu verwendet, die Abtastrichtungen zu optimieren, die Anzahl der Schattenstrahlen zu reduzieren, Caustics zu rendern und die Anzahl der Reflexionen zu begrenzen, die beim Rendern der Szene verfolgt werden. Beim Speichern eines Photons werden die folgenden vier Werte in der Photon Map abgelegt:

Erfunden von Henrik Wann Jensen ist Photon Mapping ein globales Beleuchtungsmodell. Dies erweitert Ray Tracing, vereinfacht Radiosity und ermöglicht Effekte wie Color Bleeding¹ oder Caustics² (effizienter³) zu berechnen (Bild 6.2). Die Berechnung eines Bildes wird in 2 Durchgängen durchgeführt. Im ersten Durchgang erfolgt der Aufbau sogenannter Photon Maps. Im zweiten Durchgang wird die Szene durch Monte Carlo Ray Tracing (Path Tracing) unter Zuhilfenahme der Photon Maps gerendert. Der wesentliche Vorteil der zwei Durchgänge liegt darin, dass die Berechnungen des Integrals L_r der Rendering Equation mit den im ersten Durchgang erstellten Photon Maps approximiert werden können. Dies ist im Vergleich zu herkömmlichen Path Tracing effizienter und führt zu einem ausreichenden Ergebnis. Für exakte Ergebnisse, wie beispielsweise in Filmen, ist Path Tracing direkt anzuwenden. [Jen] [Mar] [CPC] [Cau22]

- Die Position des Photons als X, Y, Z Float-Werte
- Die Farbe oder die Energie des Photons als 4 Char-Werte (als Ward's Shared-Exponent RGB-Format)
- Die Einfallsrichtung des Photons als 2 Char-Werte (Phi und Theta in sphärischen Koordinaten)
- Ein Flag-Wert als Short-Wert für den k-d Baum

Der Speicherbedarf wird durch eine möglichst effiziente Kompression der Daten minimiert. Im Anschluss an die Speicherung aller Photonen in einer Photon Map ist sicherzustellen, dass der k-d-Baum ausbalanciert ist. Dies erfolgt in der Regel nach Abschluss des ersten Durchgangs. Ein ausbalancierter k-d-Baum gewährleistet, dass das Auffinden von M Photonen in einem Baum mit N Photonen innerhalb von

¹Color Bleeding bezeichnet das Färben naher Oberflächen durch die Reflexion farbigen Lichtes [Bir02]

²Wenn Licht von spiegelnden oder reflektierenden Oberflächen reflektiert oder durch brechende Oberflächen fokussiert wird kann ein Effekt bezeichnet als Caustics entstehen. Caustics sorgen dafür, dass andere Oberflächen indirekt mit fokussierten Lichtmustern beleuchtet werden [Bir01]

³Im Gegensatz zum Radiosity Verfahren erfolgen die Berechnungen der Effekte effizienter; Im Kontrast zu Ray Tracing sind sie überhaupt erst möglich

$O(M - \log_2(N))$ Zeit erfolgt. Da sich die gesuchten Photonen in denselben Teilen des Baumes befinden, ist das Auffinden in der Praxis sogar noch effizienter.

Beim Photon Mapping werden für das Rendern von Szenen zwei Photon Maps genutzt. Die eine Photon Map ist eine Caustics Photon Map, welche nur Photonen speichert, die unmittelbar im Zusammenhang mit Caustics stehen. Die andere Photon Map ist die globale Photon Map, welche eine grobe Annäherung an die Lichtverhältnisse der Szene speichert. Dazu werden alle anderen Photonen, welche ein Objekt treffen und nicht unmittelbar im Zusammenhang mit Caustics stehen gespeichert.

Die Trennung der Photon Maps erhöht die Geschwindigkeit, die Genauigkeit und reduziert den Speicherbedarf. So ist beispielsweise das Rendern von Caustics schneller oder das Auffinden von Photonen in der globalen Photon Map (Auffinden in $O(M - \log_2(N))$ Zeit). Dies liegt dem Grunde das die Photon Maps nur Photonen mit ähnlichen Energieniveaus enthalten. So beinhaltet die Caustics Photon Map nur Photonen mit hoher Dichte und niedriger Energie, welche die Photonen sind die mit Caustics direkt in Verbindung stehen. Die globale Photon Map hingegen speichert nur normale Photonen mit niedriger Dichte und hoher Energie. [Jen] [Wat]

Die Bilder 4.1 und 4.2 zeigen die Caustics Photon Map beziehungsweise die globale Photon Map direkt visualisiert.

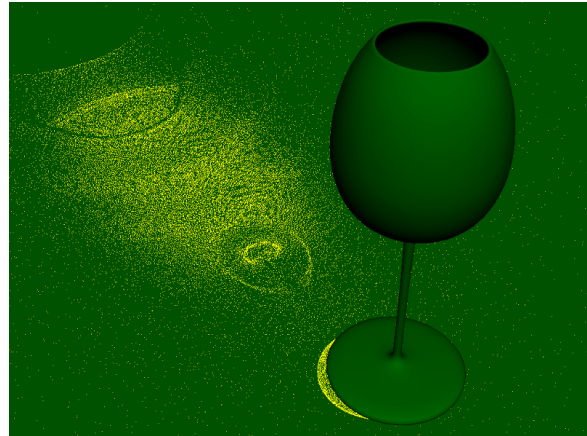


Abbildung 4.1: Die gelben Punkte repräsentieren die gespeicherten Caustics der Caustics Photon Map (Quelle: [06])



Abbildung 4.2: Direkte Visualisierung der globalen Photon Map (Quelle: [Jen])

4.2 Durchgang 1: Generieren der Photon Maps

Zunächst werden viele Photonen von den Lichtquellen aus in die Szene gesendet und verfolgt. Dieses Verfahren wird als Light

Ray Tracing bezeichnet (Abschnitt ??). Trifft ein Photon auf eine Oberfläche, so wird es in eine der beiden Photon Maps gespeichert. Ob ein Photon danach reflektiert oder absorbiert wird, wird durch Russian Roulette⁴ bestimmt. Wird es reflektiert, so wird die neue Richtung mit Hilfe der BRDF berechnet.

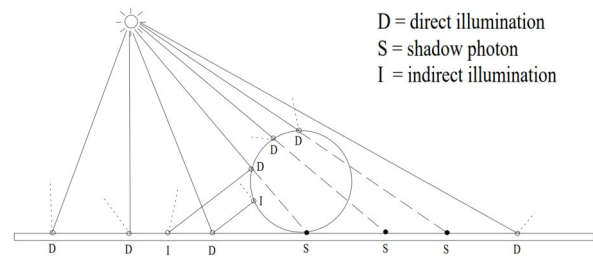


Abbildung 4.3: Klassifizierung der Photonen in der globalen Photon Map (Quelle: [Jen])

Steht ein Photon unmittelbar im Zusammenhang mit Caustics, wurde es also zuvor von spiegelnden oder transparenten Objekten reflektiert, bevor es auf eine diffuse Oberfläche trifft wird es in der Caustics Photon Map gespeichert. Die Caustics werden dann im zweiten Durchgang direkt auf Basis der Map gerendert. Aus diesem Grund muss die Map eine hohe Photondichte aufweisen.

4.3 Durchgang 2: Rendering

Alle anderen Photonen welche zuvor ein Objekt getroffen haben und nicht unmittelbar im Zusammenhang mit Caustics stehen werden in der globalen Photon Map gespeichert. Dies sind zusätzlich zu den Photonen für direktes und indirektes Licht auch sogenannte Schatten-Photonen (Shadow-Photonen). Am ersten Schnittpunkt mit einem Objekt wird ein normales Photon gespeichert, für jeden weiteren Schnittpunkt wird ein Schatten-Photon gespeichert (Bild 4.3). Die globale Photon Map muss keine so hohe Dichte wie die Caustics Photon Map aufweisen, da sie nicht direkt gerendert wird. [Lev] [Jen]

Das fertige Bild wird mit Hilfe von Monte Carlo Ray Tracing gerendert. Die Strahlungsdichte eines Pixels (Farbe eines Pixels) ist das Mittel mehrerer Stichprobenschätzungen. Um die Strahlungsdichte für einen Pixel zu bestimmen wird, wie beim Whitted-style Ray Tracing, ein Ray vom Eye aus durch jeden Pixel in die Szene gesendet und verfolgt. Trifft der Ray auf eine Oberfläche so wird die Strahlungsdichte ($L_s(x, \Psi_r)$) am Schnittpunkt x in die Richtung des Rays Ψ_r mit Hilfe der Rendering Equation berechnet.

Wie bereits erwähnt würde das direkte Lösen der Rendering Equation das beste Ergebnis erzeugen. Allerdings ist dies eben sehr rechenaufwändig, weshalb Photon Mapping das Integral L_r der Rendering Equation in eine Summe von mehreren Komponenten aufteilt:

⁴Das Monte Carlo Verfahren „Russian Roulette“ dient der Reduktion des Rechenaufwands, indem Lichtstrahlen mit geringer Energie probabilistisch terminiert werden [AK]

$$\begin{aligned}
L_r = & \int_{\Omega} f_r L_{i,l} \cos \theta_i, d\omega_i + \\
& \int_{\Omega} f_{r,s} (L_{i,c} + L_{i,d}) \cos \theta_i, d\omega_i + \\
& \int_{\Omega} f_{r,d} L_{i,c} \cos \theta_i, d\omega_i + \\
& \int_{\Omega} f_{r,d} L_{i,d} \cos \theta_i, d\omega_i
\end{aligned}$$

In der Summe wurde die BRDF f_r in die folgenden Komponenten aufgeteilt:

$f_{r,d}$	diffuser Anteil (Lambertian bis leicht glänzend)
$f_{r,s}$	spiegelnder Anteil (hochglänzend und ideal spiegelnd)

Es gilt $f_r = f_{r,d} + f_{r,s}$.

Die einfallende Strahlungsdichte L_i der Summe wurde in die folgenden Komponenten aufgeteilt:

$L_{i,l}$	Beiträge der Lichtquellen
$L_{i,c}$	Beiträge der Lichtquellen durch spiegelnde Reflexion
$L_{i,d}$	Indirekte weiche Beleuchtung

Es gilt $L_i = L_{i,l} + L_{i,c} + L_{i,d}$.

Die Auswertung der Integrale aus der Summe (Formel ??) kann auf zwei Arten erfolgen, der präzisen Berechnung oder der Approximation. Die präzise Berechnung findet Anwendung, wenn die Oberfläche direkt durch das Eye oder durch ein paar spiegelnde Reflexionen sichtbar ist oder wenn die Distanz vom Eye zum Schnittpunkt unter einem geringen Schwellwert liegt (dies verhindert in Ecken ungenaues Color Bleeding). Die Approximation wird verwendet, nachdem der Ray mindestens

einmal diffus reflektiert wurde oder die Gewichtung zu gering wird, sodass der Ray nur wenig zur Strahlungsdichte des Pixels beiträgt. [Jen]

4.3.1 Term 1: Direkte Beleuchtung

Der erste Term der Summe beschreibt den Beitrag aller Lichtquellen zur direkten Beleuchtung. Normalerweise wird dies, wie bei Raycasting oder Whitted-style Ray Tracing, durch das Aussenden von Schattenstrahlen und der Prüfung ob die Lichtquelle verdeckt wird oder nicht geprüft. Um diese Schattenstrahlen zu verhindern kann bei Anwendung der präzisen Berechnung die Beobachtung genutzt werden, dass viele Szenen große beleuchtete oder große schattige Bereiche enthalten. Trifft der Ray einen Bereich der einen Mix aus normalen direkten Photonen (direct illumination, (Bild 4.3)) und Schattenphotonen in der globalen Photon Map enthält oder wenn die Anzahl an normalen Photonen beziehungsweise Schattenphotonen zu gering ist werden Schattenstrahlen eingesetzt. Bei der Approximation wird die aus der globalen Photon Map gewonnene Strahlungsdichteschätzung verwendet.

4.3.2 Term 2: Spiegelnde Reflexionen

Das spiegelnde und hochglänzende von Oberflächen reflektierte Licht (Term 2) wird durch gewöhnliches Monte Carlo Ray Tracing berechnet. Meistens kann eine limitierte Anzahl an Rays verwendet werden,

wenn Importance Sampling⁵ basierend auf der BRDF zum Einsatz kommt.

4.3.3 Term 3: Caustics

Die Caustics auf diffusen und leicht glänzenden Oberflächen werden durch den dritten Term repräsentiert. Caustics werden nie durch Monte Carlo Stichproben berechnet, da dies in den meisten Fällen nahezu unmöglich ist, stattdessen werden sie durch die direkte Visualisierung der Caustics Photon Map dargestellt.

4.3.4 Term 4: Weiches indirektes Licht

Der vierte und finale Term der Summe umfasst jenes Licht, welches seit dem Verlassen der Lichtquelle mindestens einmal diffus reflektiert wurde. Das Licht wird wiederum durch $f_{r,d}$ auf der Oberfläche des Schnittpunktes diffus reflektiert. Die präzise Berechnung setzt Importance Sampling ein und kombiniert Daten aus der globalen Photon Map mit der BRDF, um optimierte Richtungen für die Stichproben zu erhalten. Für die Modellierung von Lambert'schen Oberflächen wird Irradiance Gradient Caching eingesetzt, wodurch die Berechnung der indirekten Beleuchtung nur dann erforderlich ist, wenn sie nicht aus zuvor berechneten Werten interpoliert werden kann. Für die Approximation wird erneut die aus der globalen Photon Map gewonnene Strahlungsdichteschätzung verwendet.

⁵Importance Sampling ist ein Verfahren zur Varianzreduzierung, welches eingesetzt werden kann um die Varianz des Approximationsfehlers zu reduzieren der bei der Monte Carlo Integration auftritt [Tab21]

5 Unterschiede & Zusammenhänge von Ray Tracing & Photon Mapping

Da nun Ray Tracing und Photon Mapping bekannt sind gibt dieses Kapitel einen Überblick über die wesentlichen Unterschiede und Zusammenhänge der beiden Verfahren.

5.1 Grundprinzip

Beim Whitted Ray Tracing werden die Rays vom Eye aus in die Szene verfolgt, während das Photon Mapping ein zweiphasiges Verfahren ist, bei dem das Light Ray Tracing zum Einsatz kommt, um Photon Maps zu erzeugen. Diese werden anschließend zum Rendern der Szene verwendet.

5.2 Beleuchtungseffekte

In Bezug auf die Beleuchtungseffekte ermöglicht Whitted Ray Tracing die Berechnung von direkter Beleuchtung, indirekter Beleuchtung in Form von spiegelnden Reflexionen, Brechungen durch transparente Materialien sowie scharfen Schat-

ten. Photon Mapping ermöglicht dagegen die Berechnung von direkter Beleuchtung, der spiegelnden und diffusen indirekten Beleuchtung (Diffuse und Specular), von Caustics, weichen Schatten und Color Bleeding.

5.3 Einschränkungen

Die Anwendung des Whitted Ray Tracings ist insofern eingeschränkt, als dass diffuse indirekte Beleuchtung sowie komplexe Lichtphänomene wie Caustics und Color Bleeding nicht effektiv dargestellt werden können. Die Berechnung dieser Phänomene ist mittels Photon Mapping zwar möglich, allerdings ist diese Methode entsprechend komplex und speicherintensiv. Infolgedessen können sowohl eine hohe Anzahl von Photonen als auch komplexe Szenarien hohe Rechenzeiten erfordern.

5.4 Anwendungsgebiete

Whitted Ray Tracing ist einfacher zu implementieren und weniger ressourcenintensiv, wodurch es sich für Szenen mit weniger komplexer Beleuchtung eignet. Es ist besonders nützlich für Anwendungen, bei denen die Berechnungsgeschwindigkeit wichtiger ist als die perfekte Genauigkeit der Lichtsimulation.

Photon Mapping hingegen ist ideal für Szenarien, in denen eine hohe visuelle Genauigkeit erforderlich ist, insbesondere bei der Darstellung von Szenen mit komplexen Lichtinteraktionen wie in Architekturvisualisierungen, wissenschaftlichen Simulationen und Filmproduktionen. Die Fähigkeit von Photon Mapping, indirekte Beleuchtung und detaillierte Lichtphänome-

ne wie Caustics und Color Bleeding zu berechnen, macht es zu einem leistungsstarken Verfahren für realistische Renderings. [\[ESK97\]](#) [\[Jen\]](#)

6 Effekte von Raytracing und Photon Mapping

In Kapitel 5 wurden die Effekte die Ray Tracing und Photon Mapping erzeugen können verglichen. In diesem Kapitel erfolgt eine grafische Darstellung der Effekte, welche mit den behandelten Verfahren erzeugt wurden. Darüber hinaus werden Abbildungen von echten Anwendungsfällen präsentiert. Die Bilder 6.1 und 6.2 zeigen Cornell-Boxen die mit den vorgestellten Verfahren gerendert wurden. An dieser Stelle sei angemerkt, dass es zu Whitted-style Ray Tracing und Photon Mapping keine aktuellen Bilder gibt, da diese Technologien mittlerweile durch optimierte Verfahren ersetzt wurden oder in diese eingeflossen sind (zum Beispiel das in Abschnitt 3.5 kurz erwähnte Verfahren Path Tracing). Dennoch wurde Ray Tracing ab 1998 in Filmen eingesetzt, wobei *A Bug's Life* der erste Film dieser Art war (Bild 6.3). Weiter ist Ray Tracing zum Beispiel in dem Film *Cars* zum Einsatz gekommen. Hier wurde eine Kombination aus Reyes¹ und vollständigem Ray Tracing verwendet. Dies sorgte für realistische spiegelnde Effekte (Specular) und detaillierte Schatten (Shadows) (Bild 6.4). Im Jahr 2006 folgte mit *Monster House* der erste mit Path Tracing

gerenderte Film (Bild 6.5). Ein neueres Beispiel zeigt eine Szene des Films *Finding Dory*, bei dem alle Effekte (Schatten, Reflexionen etc.) mit Path Tracing (RenderMan-Implementierung) gerendert wurden (Bild 6.6). Heutzutage können größere Szene, wie Bild 6.7 mit Path Tracing (Monte Carlo Ray Tracing) in **Echtzeit** gerendert werden. Bild 6.8 stellt die Ergebnisse mehrerer Verfahren einander gegenüber. Ein aktuelles Beispiel, veröffentlicht im Juli 2023, zeigt *Cyberpunk 2077* in Echtzeit gerendert mit Ray Tracing (Bild 6.9) und Path Tracing (Bild 6.10). [Whi18] [NVI22] [Cau19] [Cau22] [CJ] [RVN23]

¹REYES (Renders Everything You Ever Saw) ist ein sogenannter Scanline-Render-Algorithmus, welcher in Pixar's RenderMan implementiert ist [CJ] [Wol08]

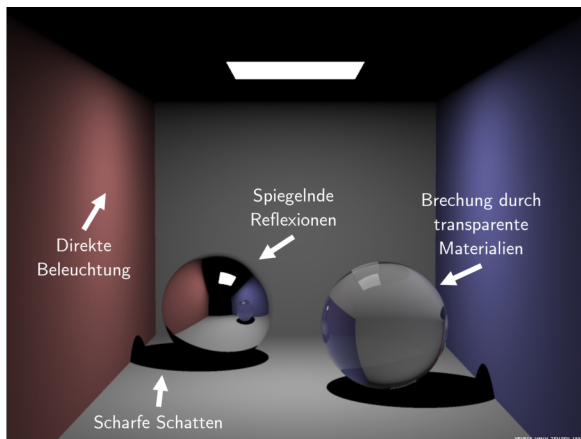


Abbildung 6.1: Cornell-Box, gerendert mit Ray Tracing (1999, Quelle: [Jen02])



Abbildung 6.3: Szene aus dem Pixar Animationsfilm *A Bug's Life* gerendert mit Ray Tracing (1998, Quelle: [CJ])

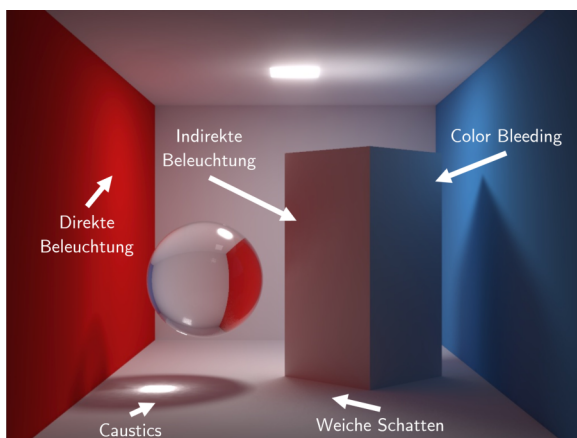


Abbildung 6.2: Cornell-Box, gerendert mit Photon Mapping (2001, Quelle: [Jen02])



Abbildung 6.4: Auto Luigi aus dem Pixar Animationsfilm *Cars* gerendert mit Reyes und Ray Tracing (2006, Quelle: [CJ])



Abbildung 6.5: Szene aus dem Columbia Pictures Industries Trickfilm *Monster House* gerendert mit Path Tracing (2006, Quelle: [CJ])

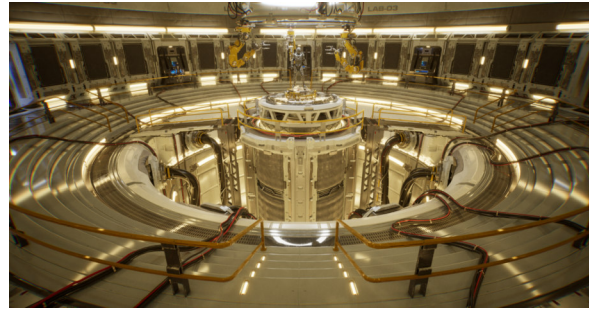


Abbildung 6.7: Demo-Szene mit Path Tracing (Monte Carlo Ray Tracing) gerendert (Quelle: [Cau22])

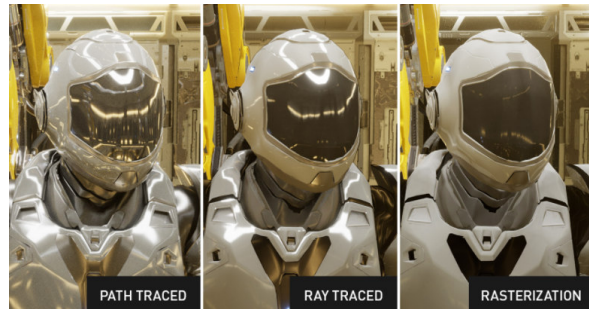


Abbildung 6.8: Raumanzug gerendert mit Path Tracing, Ray Tracing und Rasterization (Quelle: [Cau22])



Abbildung 6.6: Szene aus dem Pixar Animationsfilm *Finding Dory* gerendert mit Path Tracing (2016, Quelle: [CJ])



Abbildung 6.9: Szene aus *Cyberpunk 2077* gerendert in Echtzeit mit Ray Tracing (2023, Quelle: [RVN23])



Abbildung 6.10: Szene aus *Cyberpunk 2077*
gerendert in Echtzeit mit
Path Tracing (2023, Quelle:
[\[RVN23\]](#))

7 Entwicklungen und Zukunftsaussichten für Ray Tracing und Photon Mapping

7.1 Allgemeine Entwicklungen

Ray Tracing findet nach wie vor in diversen Bereichen Anwendung, wenngleich nicht in seiner puren Form als Whitted-style Ray Tracing. Die Anwendungsmöglichkeiten von Ray Tracing sind vielfältig. So findet es beispielsweise Verwendung in der Architekturdarstellung, beim Lichtdesign für Theater und Fernsehen sowie durch Lichtdesigner, Solarenergieforscher oder Maschinenbauingenieure. Auch immer mehr Spiele unterstützen Ray Tracing. Außerdem zeigen die Hardware-Entwicklungen der letzten Jahre das Echtzeit Path Tracing aufgrund von immer schneller werdenden Grafikprozessoren durchaus möglich ist. So wie Filme zunächst einige Ray Tracing Techniken einsetzten, bis sie komplett mit Ray Tracing gerendert wurden konnten und später mit Path Tracing, haben auch Spiele damit begonnen, Ray Tracing in begrenztem Umfang zu nutzen. Mittlerweile gibt

es eine wachsende Zahl von Spielen die teilweise Ray Tracing einsetzen. So werden traditionelle auf Rasterung basierende Rendering Verfahren mit einigen Ray Tracing Effekten kombiniert. Daher ist davon auszugehen das Ray Tracing durch Path Tracing oder möglicherweise sogar durch (Progressive) Photon Mapping abgelöst wird. Es ist bereits gelungen nicht nur Path Tracing als Echtzeit Rendering Verfahren einzusetzen, sondern sogar schon Photon Mapping. Für ausgewählte Demo-Szenen die mit Photon Mapping gerendert wurden, wurden so Rendering Zeiten von 14,2 bis 21,7 Milisekunden erreicht. Dennoch ist zu berücksichtigen, das es durchaus noch ein weiterer Weg ist damit Path Tracing und Photon Mapping die bisherigen Rendering Verfahren vollständig ablösen, da die Hardware-Nutzung noch sehr durchwachsen ist zwischen Hardware die spezielle Rechenkerne für Ray Tracing etc. bereitstellt und Hardware vor dieser Entwicklung die entsprechend Ray Tracing etc. nur langsamer berechnen kann. [HOJ] [Cau22] [NV1b] [SA19] [Val24]

7.2 Ray Tracing

NVIDIA entwickelt mit Subwarp Interleaving ein neues Verfahren, welches eine Erhöhung der Hardwareauslastung sowie eine Reduktion der Warp-Latenz¹ ermöglicht. Die Leistungssteigerung beträgt bislang 6,3% und in manchen Fällen sogar bis zu 20%. Allerdings sind architektonische Änderungen erforderlich, um das Verfahren nutzbar zu machen. Daher ist anzunehmen, dass frühestens die 5000er Serie dieses Verfahren implementiert bekommt.

¹Warps sind eine nicht physische Organisationsstruktur innerhalb von NVIDIA-Grafikkarten, die verwendet wird, um (32) Threads in Gruppen zu organisieren und effizient parallel auszuführen [Spr23]

Des Weiteren optimiert NVIDIA die Grafikqualität in diversen Spielen durch die Implementierung von DLSS 3.5 mit DLSS Ray Reconstruction. Bei Aktivierung ersetzt DLSS Ray Reconstruction handoptimierte Ray Tracing Denoiser durch ein neues, einheitliches KI-Modell. Diese speziellen Ray Tracing Funktionen werden durch die Ausstattung der RTX-Grafikkarten mit RT-Cores ermöglicht.

Auch AMD hat die Ray Tracing Technologie in ihre Radeon-RX-Grafikkarten mit der RDNA3-Architektur integriert, wodurch sowohl Ray Tracing als auch KI-Beschleunigung ermöglicht wird.

Inzwischen stehen mit DirectX Ray Tracing (DXR) und Vulkan auch API-Implementierungen von Ray Tracing zur Verfügung. [Dam+] [Cor22] [NVIa] [AMD] [D3D18] [The20] [Sta]

anderen Monte Carlo Ray Tracing Verfahren (Bild 7.1). [HOJ]

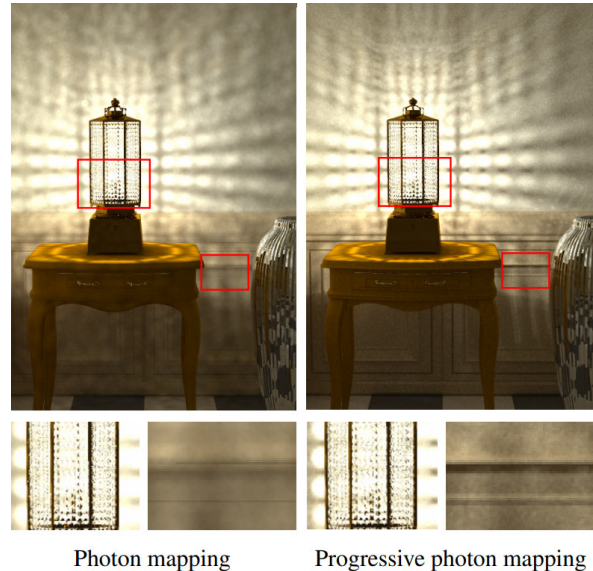


Abbildung 7.1: Vergleich zwischen Photon Mapping und Progressive Photon Mapping (Quelle: [HOJ])

7.3 Photon Mapping

Mit Progressive Photon Mapping (PPM) existiert bereits eine Verbesserung des ursprünglichen Photon Mapping Verfahrens. Beim PPM wird die Reihenfolge und die Anzahl der Durchgänge verändert. So wird im ersten Durchgang Eye Ray Tracing verwendet, während beliebig viele darauffolgende Durchgänge Photon Tracing durchführen. Dieser Ansatz bietet die Vorteile, dass weniger Speicher benötigt wird, da im Gegensatz zum Photon Mapping keine vollständigen Photon Maps gespeichert werden müssen, sowie dass die Genauigkeit steigt, umso mehr Photon Tracing Durchgänge ausgeführt werden. PPM ist bei komplexem Licht, wie es bei Caustics oder der Reflektion von Caustics auftritt, ein robustes und effizientes Verfahren verglichen mit

Literatur

- [06] M. *Caustics*. 28. Jan. 2006. URL: <https://de.m.wikipedia.org/wiki/Datei:Caustics.png>.
- [AK] J. A. Arvo und D. K. Kirk. “Particle Transport and Image Synthesis”. DOI: [10.1145/97879.97886](https://doi.org/10.1145/97879.97886).
- [AMD] AMD. *AMD Radeon™ RX Grafikkarten*. URL: <https://www.amd.com/de/products/graphics/desktops/radeon.html> (besucht am 13.06.2024).
- [Bir01] J. B. Birn. *Caustics*. 2001. URL: <https://www.3drender.com/glossary/caustics.htm> (besucht am 17.07.2024).
- [Bir02] J. B. Birn. *Color Bleeding*. 2002. URL: <https://www.3drender.com/glossary/colorbleeding.htm> (besucht am 13.06.2024).
- [Cau19] B. Caulfield. *What’s the Difference Between Hardware- and Software-Accelerated Ray Tracing?* / *NVIDIA Blog*. 7. Juni 2019. URL: <https://blogs.nvidia.com/blog/whats-the-difference-between-hardware-and-software-accelerated-ray-tracing/> (besucht am 14.07.2024).
- [Cau22] B. Caulfield. *What Is Path Tracing?* / *NVIDIA Blog*. 23. März 2022. URL: <https://blogs.nvidia.com/blog/what-is-path-tracing/> (besucht am 14.07.2024).
- [CJ] P. H. C. Christensen und W. J. Jarosz. “The Path to Path-Traced Movies”. DOI: [10.1561/06000000073](https://doi.org/10.1561/06000000073).
- [Cor22] H. Corrigan. *Future Nvidia GPUs could be 20% better at Ray Tracing*. 27. Jan. 2022. URL: <https://www.pcgamer.com/future-nvidia-gpus-could-be-20-better-at-ray-tracing/> (besucht am 12.06.2024).
- [CPC] R. L. C. Cook, T. P. Porter und L. C. Carpenter. “Distributed Ray Tracing”.
- [D3D18] D3D Team. *Announcing Microsoft DirectX Raytracing!* 19. März 2018. URL: <https://devblogs.microsoft.com/directx/announcing-microsoft-directx-raytracing/> (besucht am 13.06.2024).
- [Dam+] S. D. Damani, M. S. Stephenson, R. R. Rangan, D. J. Johnson, R. K. Kulkarni und S. W. K. Keckler. “GPU Subwarp Interleaving”. DOI: [10.1109/HPCA53966.2022.00090](https://doi.org/10.1109/HPCA53966.2022.00090).
- [ESK97] J. L. Encarnação, W. Straßer und R. Klein. *Graphische Datenverarbeitung 2. Modellierung komplexer Objekte und photorealistische Bilderzeugung*. 4. Aufl. De Gruyter Oldenbourg, 1997.

- [Fre] Freie Universität Berlin. *Z – Buffer Algorithmus*. URL: https://www.inf.fu-berlin.de/lehre/WS06/19605_Computergrafik/doku/rossbach_siewert/zbuffer.html (besucht am 11.07.2024).
- [Gla19] A. S. G. Glassner. *An Introduction to Ray Tracing*. ACADEMIC PRESS LIMITED, 11. Apr. 2019. URL: <https://www.realtimerendering.com/raytracing/An-Introduction-to-Ray-Tracing-The-Morgan-Kaufmann-Series-in-Computer-Graphics-.pdf>.
- [Gri08] C. Gribble. Ray Tracing For Undergraduates. In: Juni 2008, S. 13.1019.1–13.1019.12. DOI: [10.18260/1-2--4066](https://doi.org/10.18260/1-2--4066).
- [HOJ] T. H. Hachisuka, S. O. Ogaki und H. W. J. Jensen. “Progressive Photon Mapping”.
- [HV] C. H. Hery und R. V. Villemin. “Physically Based Lighting at Pixar”.
- [Hv] J. F. H. Hughes und A. V. D. van Dam. *Introduction to Radiosity*. URL: <https://www.eecs.umich.edu/courses/eecs487/W04/Lectures/radiosity.pdf>.
- [Jen] H. W. J. Jensen. “Global Illumination using Photon Maps”.
- [Jen02] H. W. J. Jensen. *Cornell box images by Henrik Wann Jensen*. 11. Nov. 2002. URL: <https://graphics.stanford.edu/~henrik/images/cbox.html> (besucht am 12.06.2024).
- [Kaj] J. T. K. Kajiya. “THE RENDERING EQUATION”. DOI: [10.1145/15886.15902](https://doi.org/10.1145/15886.15902).
- [Kro+] D. P. K. Kroese, T. B. Brereton, T. T. Taimre und Z. I. B. Botev. “Why the Monte Carlo method is so important today”. DOI: [10.1002/wics.1314](https://doi.org/10.1002/wics.1314).
- [Lev] D. L. Leventhal. *Ray Tracing*. URL: <https://courses.cs.washington.edu/courses/cse457/11au/lectures/markup/ray-tracing-markup.pdf>.
- [LGK] K. Lehn, M. Gotzes und F. Klawonn. *Grundlagen der Computergrafik. Eine Einführung mit OpenGL und Java*. 4. Aufl. Springer Vieweg. DOI: [10.1007/978-3-658-36075-7](https://doi.org/10.1007/978-3-658-36075-7). URL: <https://link.springer.com/book/10.1007/978-3-658-36075-7>.
- [Mar] A. M. Martin. *Distributed Ray Tracing*. URL: https://web.cs.wpi.edu/~matt/courses/cs563/talks/dist_ray/dist.html (besucht am 13.06.2024).
- [Mas07] T. M. Masson. *CG 101 : a computer graphics industry reference*. 2. Aufl. Digital Fauxtography, 2007. URL: https://archive.org/details/isbn_9780977871001/page/268/mode/2up.
- [Moh23] L. M. Mohrmann. *Computer graphics and interaction, Lecture 10*. 2023. URL: https://hexagon-sphere.com/lectures/cgi_lecture_10.pdf.

- [Nic+77] F. E. N. Nicodemus, J. C. R. Richmond, J. J. H. Hsia, I. W. G. Ginsberg und T. L. Limperis. *Geometrical Considerations and Nomenclature for Reflectance*. NATIONAL BUREAU OF STANDARDS, Okt. 1977.
- [NVIa] NVIDIA. *Unbekannt*. URL: <https://www.nvidia.com/en-us/geforce/news/gfecnt/20241/ces-2024-dlss-ray-tracing-rtx-games-announcements/> (besucht am 12.06.2024).
- [NVIb] NVIDIA Corporation. *NVIDIA RTX Path Tracing*. URL: <https://developer.nvidia.com/rtx/path-tracing> (besucht am 15.07.2024).
- [NVI22] NVIDIA. *NVIDIA on LinkedIn: What Is Path Tracing? | NVIDIA Blog*. 2022. URL: https://www.linkedin.com/posts/nvidia_what-is-path-tracing-nvidia-blog-activity-6921830092398592000-yLJt (besucht am 14.07.2024).
- [RVN23] P. Reuther, R. Vötter und D. Ney. *Cyberpunk 2077 mit Pathtracing: GPUs fallen bei echter Next-Gen-Grafik um wie die Fliegen [Update mit RTX 4060]*. 2. Juli 2023. URL: <https://www.pcgameshardware.de/Cyberpunk-2077-Spiel-20697/Specials/Overdrive-Pathtracing-Update-Benchmark-1418393/> (besucht am 15.07.2024).
- [SA19] N. Smal und M. Aizenshtein. Real-Time Global Illumination with Photon Mapping. In: *Ray Tracing Gems: High-Quality and Real-Time Rendering with DXR and Other APIs*. Hrsg. von E. Haines und T. Akenine-Möller. Berkeley, CA: Apress, 2019, S. 409–436. ISBN: 978-1-4842-4427-2. DOI: [10.1007/978-1-4842-4427-2_24](https://doi.org/10.1007/978-1-4842-4427-2_24). URL: https://doi.org/10.1007/978-1-4842-4427-2_24.
- [Spr23] F. S. Sprengel. *Visual Computing – GPU Computing, Threads and Memory*. 2023.
- [Sta] Stanford University. *Applications of Ray Tracing*. URL: <https://cs.stanford.edu/people/eroberts/courses/soco/projects/ray-tracing/applications.html> (besucht am 13.06.2024).
- [Tab21] M. T. Taboga. *Importance sampling | Explanation, formulae, example*. 2021. URL: <https://www.statlect.com/asymptotic-theory/importance-sampling> (besucht am 17.07.2024).
- [The] The University of Texas at Austin. *Basic Ray Tracing*. URL: <https://www.cs.utexas.edu/~theshark/courses/cs354/lectures/cs354-4.pdf>.
- [The20] The Khronos Group Inc. *Ray Tracing In Vulkan*. 15. Dez. 2020. URL: <https://www.khronos.org/blog/ray-tracing-in-vulkan> (besucht am 13.06.2024).
- [Uni] Universität Leipzig. *Beleuchtung, COMPUTERGRAPHIK*. URL: <https://www.informatik.uni-leipzig.de/bsv/homepage/sites/default/files/CG-9-Beleuchtungsmodelle.pdf>.
- [Val24] Valve Corporation. *Steam Hardware Software Survey*. Juni 2024. URL: <https://store.steampowered.com/hwsurvey/videocard/> (besucht am 15.07.2024).

- [Wat] Z. W. Waters. *Photon Mapping - Zack Waters*. URL: https://web.cs.wpi.edu/~emmanuel/courses/cs563/write_ups/zackw/photon_mapping/PhotonMapping.html (besucht am 17.07.2024).
- [Whi] T. W. Whitted. “An Improved Illumination Model for Shaded Display”. DOI: [10.1145/358876.358882](https://doi.org/10.1145/358876.358882).
- [Whi18] J. T. W. Whitted. *A Ray-Tracing Pioneer Explains How He Stumbled into Global Illumination*. 1. Aug. 2018. URL: <https://blogs.nvidia.com/blog/ray-tracing-global-illumination-turner-whitted/> (besucht am 12.06.2024).
- [Wik04] Wikipedia-Autoren. *Phong-Beleuchtungsmodell*. 27. Feb. 2004. URL: <https://de.wikipedia.org/wiki/Phong-Beleuchtungsmodell> (besucht am 12.06.2024).
- [Wol08] E. W. Wolff. *RenderMan@20: Ed Catmull and Dana Batali Reflect On Pixar’s Killer App. Two of Pixar’s principals take a look at the past, present and future of the landmark rendering system*. 13. Aug. 2008. URL: <https://www.awn.com/vfxworld/renderman20-ed-catmull-and-dana-batali-reflect-pixars-killer-app> (besucht am 15.07.2024).